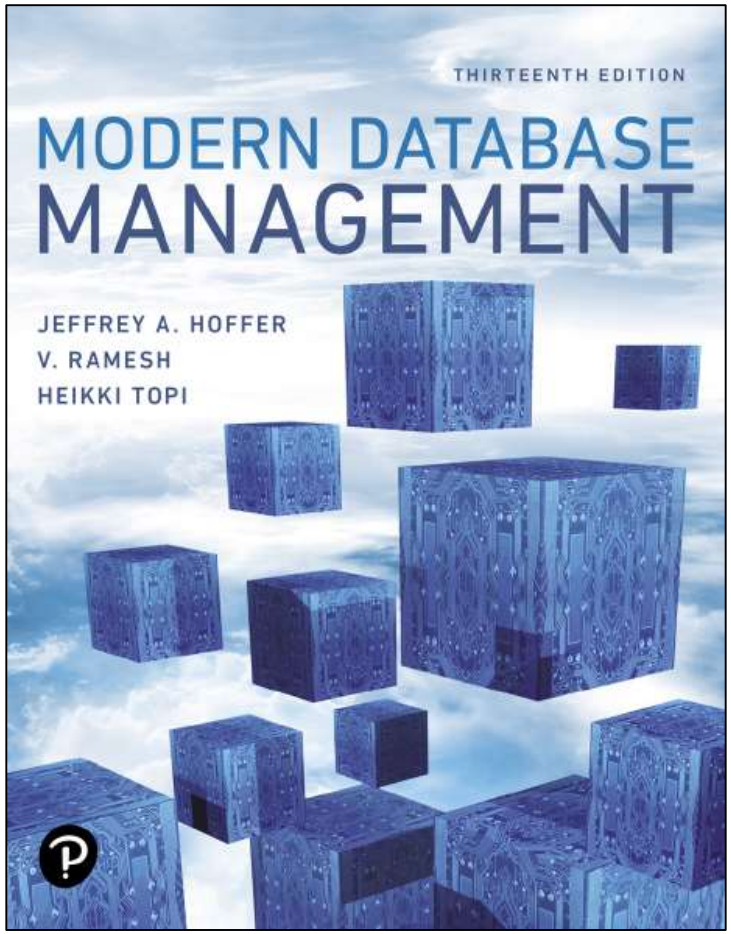
Modern Database Management

Thirteenth Edition



Chapter 4

Logical Database Design and the Relational Model

Data Normalization

- Primarily a tool to validate and improve a logical design so that it satisfies certain constraints that **avoid unnecessary duplication of data**
- The process of decomposing relations with anomalies to produce smaller, **well-structured relations**

Well-Structured Relations

- Relations that contain minimal data redundancy and allow users to insert, delete, and update rows without causing data inconsistencies
- Goal is to avoid anomalies
  - **Insertion Anomaly** – adding new rows forces user to create duplicate data
  - **Deletion Anomaly** – deleting rows may cause a loss of data that would be needed for other future rows
  - **Modification Anomaly** – changing data in a row forces changes to other rows because of duplication

Example–Figure 4-2b

EMPLOYEE2

EmpID	Name	DeptName	Salary	CourseTitle	DateCompleted
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/2018
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/2018
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/2018
110	Chris Lucero	Info Systems	43,000	Visual Basic	1/12/2018
110	Chris Lucero	Info Systems	43,000	C++	4/22/2018
190	Lorenzo Davis	Finance	55,000		
150	Susan Martin	Marketing	42,000	SPSS	6/19/2018
150	Susan Martin	Marketing	42,000	Java	8/12/2018

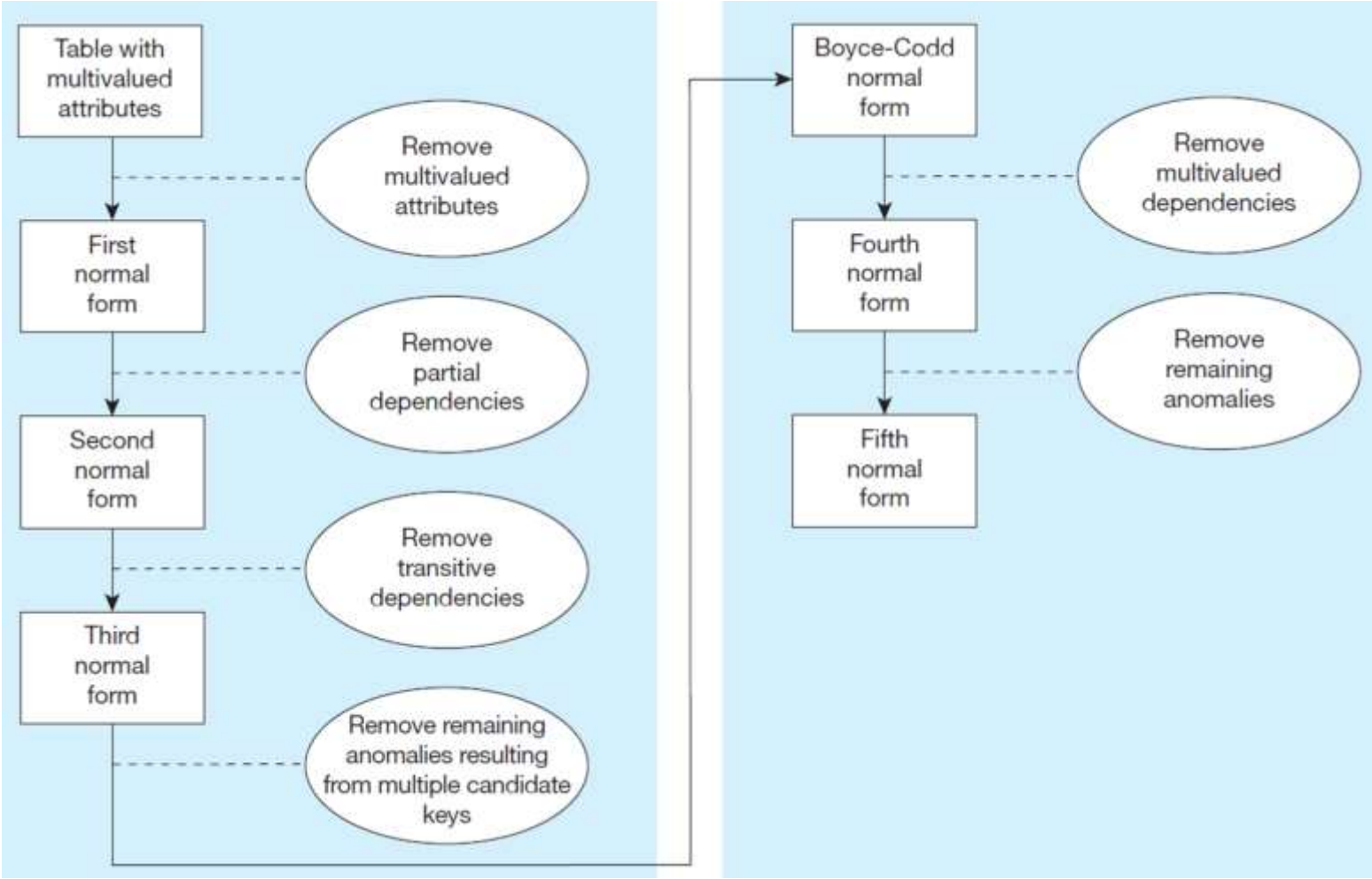
Question: Is this a relation? Answer: Yes; unique rows and no multivalued attributes

Question: What's the primary key? Answer: Composite — EmpID, CourseTitle

Anomalies in This Relation (1 of 2)

- **Insertion** – can't enter a new employee without having the employee take a class (or at least empty fields of class information)
- **Deletion** – if we remove employee 140, we lose information about the existence of a Tax Acc class
- **Modification** – giving a salary increase to employee 100 forces us to update multiple records

Figure 4.22 Steps in Normalization



Functional Dependencies and Keys

- Functional Dependency: The value of one attribute (the **determinant**) determines the value of another attribute
- Candidate Key:
  - A unique identifier. One of the candidate keys will become the primary key
    - E.g., perhaps there is both credit card number and SS# in a table…in this case both are candidate keys.
  - Each non-key field is functionally dependent on every candidate key.

First Normal Form

- No multivalued attributes
- Every attribute value is atomic
- Fig. 4-25 –next slide- **is not** in 1st Normal Form (multivalued attributes) → it is not a relation.
- Fig. 4-26 *is* in 1st Normal form.
- **All relations are in 1st Normal Form.**

Figure 4.25 Invoice Data (Pine Valley Furniture Company)

OrderID	Order Date	Customer ID	Customer Name	Customer Address	Product ID	Product Description	Product Finish	Product StandardPrice	Ordered Quantity
1006	10/24/2018	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
					5	Writer's Desk	Cherry	325.00	2
					4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2018	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
					4	Entertainment Center	Natural Maple	650.00	3

Table with multivalued attributes, not in 1st normal form.

This is **not** a relation.

Figure 4.26 INVOICE Relation (1NF) (Pine Valley Furniture Company)

OrderID	Order Date	Customer ID	Customer Name	Customer Address	ProductID	Product Description	Product Finish	Product Standard Price	Ordered Quantity
1006	10/24/2018	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2018	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2018	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2018	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
1007	10/25/2018	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

This is a relation, but not a well-structured one.

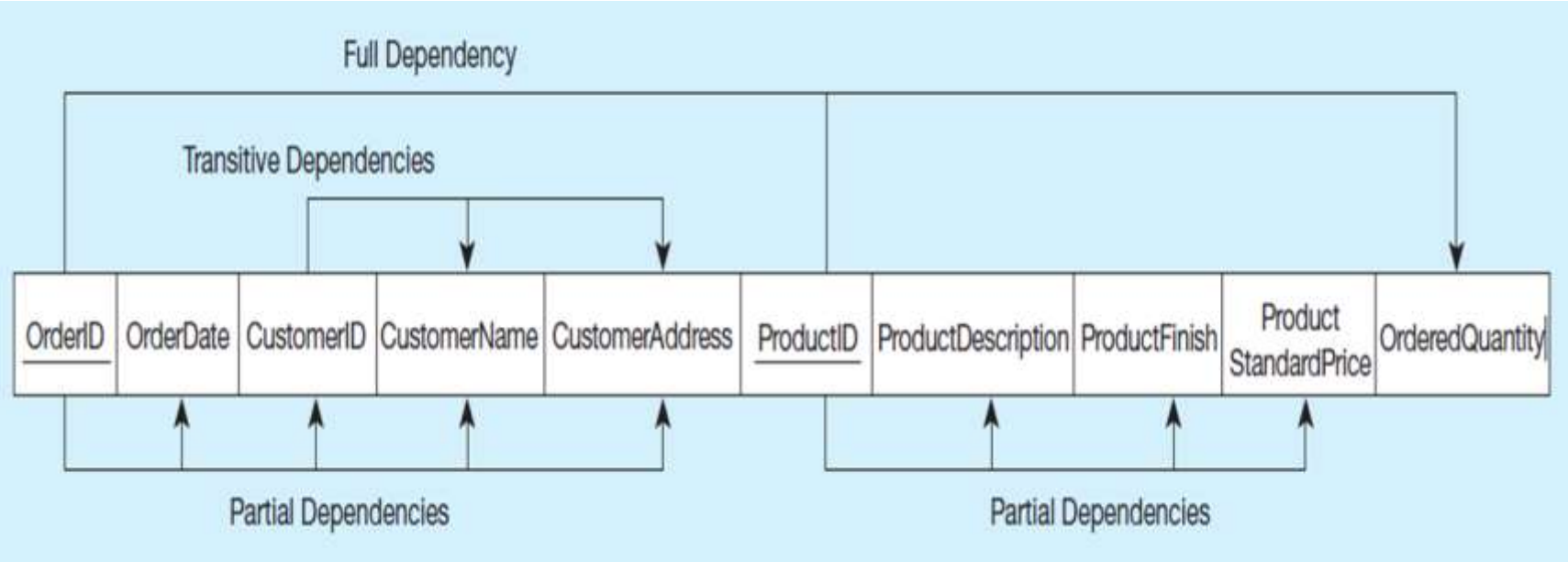
Anomalies in This Relation (2 of 2)

- **Insertion** – if new product is ordered for order 1007 of existing customer, customer data must be re-entered, causing duplication
- **Deletion** – if we delete the Dining Table from Order 1006, we lose information concerning this item's finish and price
- **Update** – changing the price of product ID 4 requires update in multiple records

Second Normal Form

- 1NF plus **every non-key attribute is fully functionally dependent on the ENTIRE primary key**
  - Every non-key attribute must be defined by the entire key, not by only part of the key
  - No partial functional dependencies

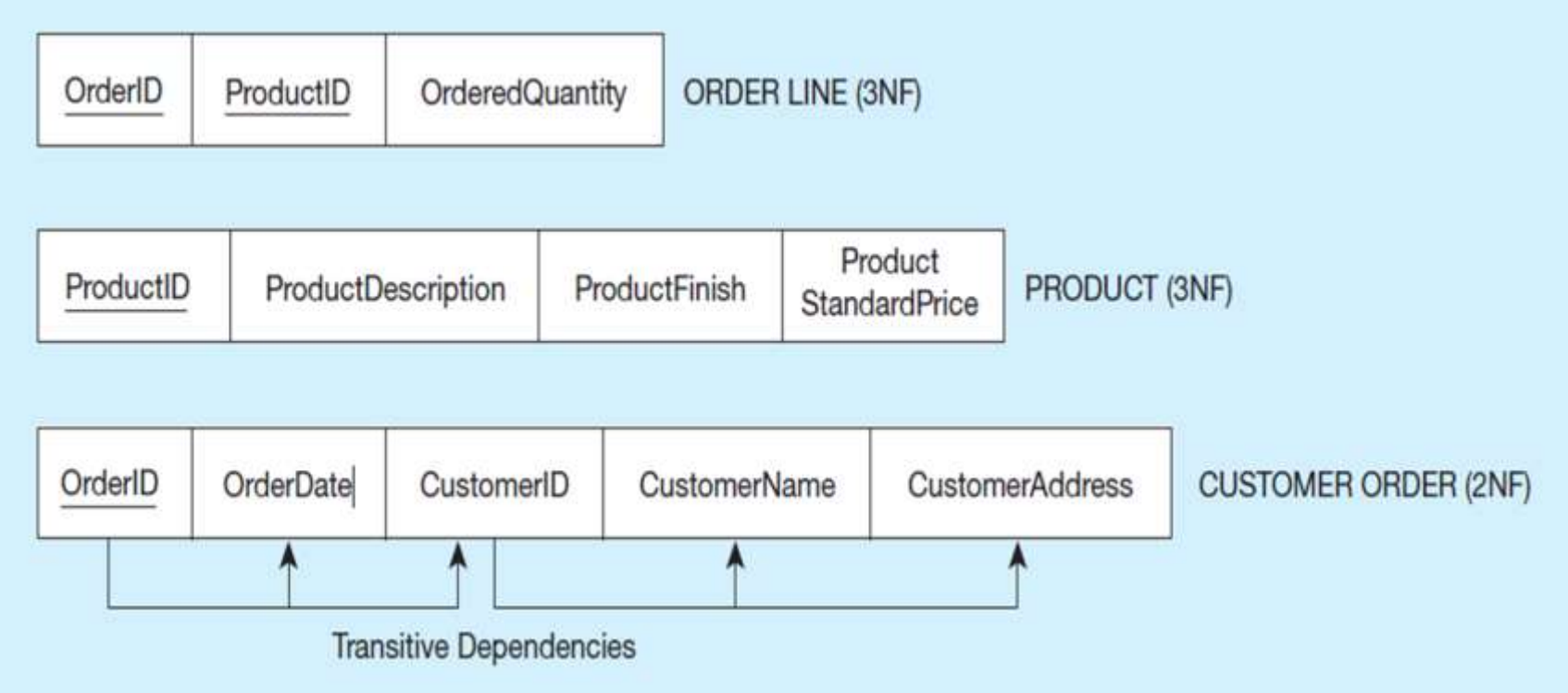
Figure 4-27 Functional Dependency Diagram for Invoice



- OrderID → OrderDate, CustomerID, CustomerName, CustomerAddress
- CustomerID → CustomerName, CustomerAddress
- ProductID → ProductDescription, ProductFinish, ProductStandardPrice
- OrderID, ProductID → OrderQuantity

Therefore, **not** in 2nd Normal Form

Figure 4-28 Removing Partial Dependencies



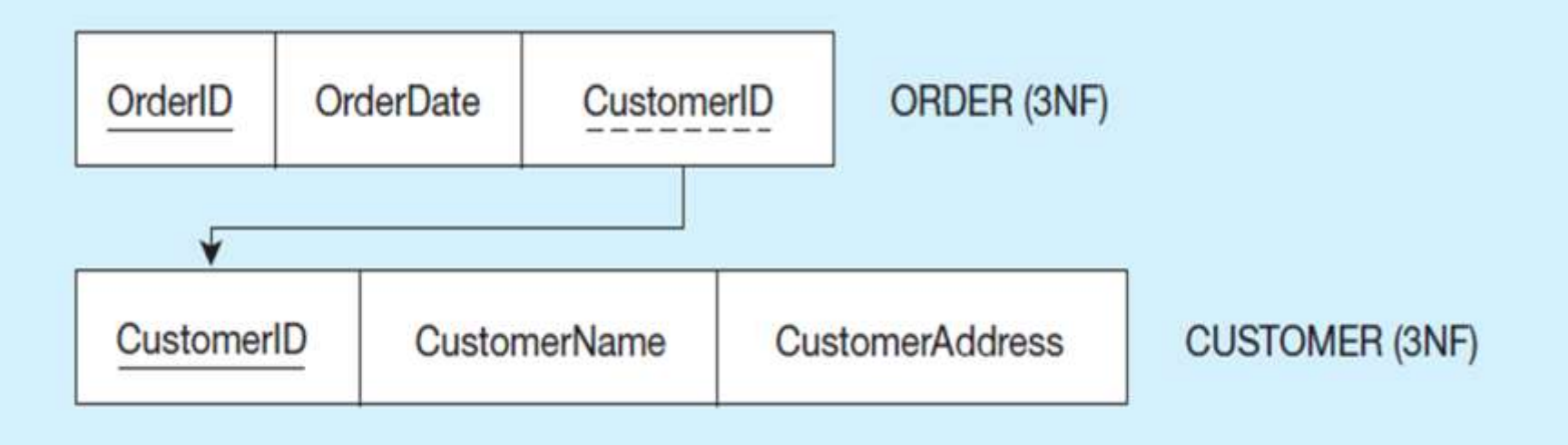
Getting it into Second Normal Form

Partial dependencies are removed, but there are still transitive dependencies

Third Normal Form

- 2NF PLUS **no transitive dependencies** (functional dependencies on non-primary-key attributes)
- Note: This is called transitive, because the primary key is a determinant for another attribute, which in turn is a determinant for a third
- Solution: Non-key determinant with transitive dependencies go into a new table; non-key determinant becomes primary key in the new table and stays as foreign key in the old table

Figure 4-29 Removing Transitive Dependencies



Getting it into Third Normal Form

Transitive dependencies are removed.

Copyright